Abstract

Discrete event simulation (DES) is a form of computer-based modelling that provides an intuitive and flexible approach to representing complex systems. It has been used in a wide range of health care applications. Most early applications involved analyses of systems with constrained resources, where the general aim was to improve the organisation of delivered services. More recently, DES has increasingly been applied to evaluate specific technologies (as a framework for health technology assessments).

The aim of this paper is to provide consensus-based guidelines on the application of DES in a healthcare setting, covering the many forms of analysis to which DES can be applied.

The paper works through the different stages of the modelling process: structural development, parameter estimation, model implementation, model analysis, and representation and reporting. At each stage, a brief description of the relevant DES processes are provided, followed by consideration of issues that are of particular relevance to the application of DES in a healthcare setting.

Each section contains a number of recommendations that were iterated amongst the authors, as well as the wider modelling taskforce, jointly set up by the International Society for Medical Decision Making and the Society for Medical Decision Making.
Introduction

Discrete event simulation (DES) is a flexible modeling method characterized by the ability to represent complex behaviour within, and interactions between individuals, populations, and their environment.[1] For example, in healthcare, this means that within the same framework both changes in health state within an individual and how that individual interacts with other individuals or with the health care system and general environment can be modeled. This flexibility allows DES to be used over a very wide range of problems.

DES was originally developed in the 1960s in the fields of industrial engineering and operations research to help analyse and improve industrial and business processes. Applications in healthcare have increased over the last 40 years,[2] and include applications relating to biologic models and physiology, process redesign and optimization, geographic allocation of resources, trial design, policy evaluation, and survival modelling, and also in health technology assessments (HTA).

All DES models define an environment that is to be simulated. This is known as the system, which may be a specific geographical location (e.g. a hospital, or a hospital department) or more general, such as a particular disease in a defined patient population (e.g. persons with cardiovascular disease in Australia).

The core concepts of DES are entities, attributes, events, resources, queues and time. Entities are the objects that can experience the events that define the model structure. In health care models, they are typically patients, but they can be other people such as caregivers or items like organs, or even signals (e.g. an email) that can interact with other entities or the system itself. Entities can be created at the start of the model or whenever it is appropriate to the problem (e.g. when the disease at issue occurs).

Attributes are features that are specific to each entity. These attributes allow each entity to “carry” information with it (like in a journal or backpack), describing, for example, its age, sex, race, current health state, past events and health states, quality of life, accumulated costs, and so on. The values assigned to different attributes may be used to determine how an entity responds to a given set of circumstances, for example, the timing and type of past events may influence the likelihood and timing of subsequent events. Attribute values may be modified at any time during the simulation. They may also be aggregated with the values of other entities and/or analysed further outside the simulation (e.g. to estimate mean cost and effects values).

Events are the central concept of a DES and are broadly defined as things that can happen to an entity (or to the environment) during the simulation. An event can be the occurrence of clinical conditions
such as the onset of an acute condition (e.g. a stroke), an adverse drug reaction, or the progression of a
disease to a new stage. Resource use (e.g. admission to hospital), clinical decisions (e.g. a change in
dose) or even experiences outside of health care (e.g. failure to show up at work) can all be represented
as events. Events can occur, and recur, in any logical sequence.

A resource is an object that provides a service to an entity, which may take the form of land, labour or
capital (equipment). Providing a service may require time. The number of entities a resource can serve
simultaneously is the resource’s capacity, for example, a clinic with a single doctor can serve one person
at a time. A clinic with three doctors can serve up to three patients simultaneously (if there is space for
the three doctors to work simultaneously). This allows the DES framework to capture spatial as well as
temporal relationships.

If a resource is occupied when an entity seeks its use, then that entity must wait until the resource is
free. This is when queues form. A queue is a place in which an entity waits for access to a needed
resource, for example, patients waiting to be called within a clinic, or a holding function, such as
patients waiting for the date of their scheduled clinic appointment. Queues have their own logic: it can
be first-in-first-out where the order of arrival in the queue is respected (e.g. a typical waiting room
queue), last-in-first-out where entities get picked from the back of the queue (like people in an
aeroplane) or based on some level of need or priority (like the ER triage).

A fundamental component of a DES is time itself. An explicit simulation clock (initiated at the start of the
model run) keeps track of the passage of time. Referencing this clock makes it possible to track interim
periods such as the length of stay in hospital, the time spent with symptoms, and, of course, the survival
(quality-adjusted if appropriate) of patients. Time can be efficiently advanced to the next interesting
time point when the next event happens, without wasting effort in unnecessary interim computations
(a patient might have nothing happening for two years and then an MI occurs, with ambulance, ER,
treatment, stroke and other multiple events occurring within minutes).

Other important concepts include interaction, which occurs whenever an entity competes with another
over a resource (e.g. entities compete for priority in a queue), and emergent behaviour, which is
behaviour that is characteristic of the system as a whole, such as spontaneous overcrowding in
emergency rooms because elective surgeries are only scheduled once a week. A strategy is an
alternative policy or configuration of the system, where the purpose of the model is to compare
different strategies in order to identify the optimal strategy.
DES models can be used to address a wide range of questions.[3,4] In this paper, we distinguish between two main categories of DES applications in health care: non-constrained-resource and constrained-resource models. Although unrealistic in health care, non-constrained-resource models accord with the common assumption made in most evaluations of the economic efficiency of alternative health care technologies: that all resources are available as needed with no capacity constraints.

In this area of application, DES allows for very flexible management of time because it handles time continuously rather than restricting occurrences to fixed time intervals.[5] DES is a particularly good choice when patients are subject to multiple risks (i.e. “competing” risks) because its continuous treatment of time allows for derivation and use of data describing the time to each event, and thus, proper application of the competing risks. Although this can be approximated in state transition models by using very short cycle lengths, this can lead to increased running times because the model has to check whether each event has occurred during every model cycle.

DES also facilitates tractable representations of complex disease pathways by embedding much of the details of the system within the individual entities (e.g. patients) rather than requiring the specification of complex model structures. Thus, DES is also a good choice when many patient characteristics must be taken into account, particularly if characteristics change over time; when what happens next depends on what happened before; when the effects of decisions made along the way (rather than only at the start) are of interest; and whenever health care or disease processes involve a series of associated events (e.g. myocardial infarct → resuscitation→ PCI stenting→ stroke).

Constrained-resource models, in addition to the above, represent interactions between individuals, for example, where multiple entities may compete for access to a limited healthcare resource, such as organs for transplant or a doctor. Constrained-resource problems generally involve entities (e.g. patients) competing for access to resources (e.g. for clinic appointments) and waiting in queues. Patients’ demand for particular resources and their priority status in a queue may be influenced by the attributes of the entity (e.g. patient characteristics or medical history). Such scenarios are the type of problem for which DES was developed, and thus, it is clearly the most appropriate choice of modeling method.

DES can also be used to model more complex interactions between individuals (e.g. transmission of disease), leading to the use of agent-based modelling (agent=entity with embedded decision logic), which is an extended application of DES that provides more detailed representation of the interactions between agents (see the dynamic transmission modelling paper for more details).
This remainder of this paper is structured around the key stages in the development, implementation, and analysis of a DES-based evaluation. These stages cover the design and structuring of a DES; the estimation and specification of input parameter values; the implementation of a DES; the process of running and analysing the implemented model; issues specific to the validation and reporting of DES models. Each section will provide a short summary of the relevant issues as they relate to DES, followed by recommendations and accompanying rationale to provides the reader with guidance about best practice in the use of DES in a health care context.

**Model structure and design**

Conceptually, DES model design starts by defining the system to be represented, followed by consideration of relevant events that can occur within the system. Events need not be restricted to those that result in an entity changing health status (e.g. a patient becoming ill), but can represent events that alter the likelihood of alternative outcomes (e.g. reperfusion following admission for myocardial infarction).

A key issue concerns the representation of disease progression, and related events (e.g. initiation of treatment). In most cases, disease progression can be represented as an event, for example, the occurrence of a relapse or a bone fracture. However, disease progression can also be represented as a continuous variable, for example, HbA1c levels in diabetes patients. Continuous measures of disease progression can be monitored using attributes, which can be updated during the course of the model.

The time spent in different health states can be specified as attributes for individual patients to facilitate the estimation of costs and quality of life effects, though such information may also influence the likelihood of subsequent events. The specification of other attributes can be informed by the events included in the model, for example, patient characteristics that influence the likelihood, severity, priority ordering, and/or outcomes of the specified events.

*Health-related outcomes are important...*

Common outcomes for constrained resource models include flow times, wait times, throughput and resource utilisation (costs). However, changes in these variables may also affect health outcomes (morbidity and/or mortality) via changes in access to care, and time to treatment. In order to fully inform decisions regarding the value of alternative systems it is important to include relevant health outcomes.

**Recommendation:**
Constrained resource models should consider health-related outcomes, and not focus solely on measures of throughput.

The need to model constrained resources should be carefully considered...

Other issues to consider include the identification of events for which representation of constrained resources is relevant, i.e. events for which entities may not have immediate access, and for which they queue. Most commonly, constrained resource events are represented in models that evaluate alternative approaches to organising the delivery of services pathways using established technologies, though there are examples of DES models that have included constrained resource events to evaluate alternative health technologies.[6]

Recommendation:

The effects of constrained resources should be modelled if:

- evaluated technologies result in differing levels of access (e.g. different referral rates), and
- time to access referred events can have significant effects on costs and/or outcomes (e.g. surgery)

Downstream decisions...

The conceptualisation of the system to be modelled should also identify decision points, for example, points at which treatment decisions are made. At each decision point, the analyst should consider whether the probabilities of alternative decisions should be represented (i.e. the likelihood of alternative downstream decisions are parameters to be estimated), or whether the analysis seeks to evaluate combinations of decisions. The latter is clearly relevant to evaluating the organisation of existing services. However, it is also potentially relevant to the evaluation of new technologies, for example, the cost-effectiveness of screening may be greatly affected by the choice of diagnostic and/or treatment decisions.

Recommendation:

If downstream decisions can have significant effects on the differences in costs and/or outcomes of the primary object of an evaluation, the model should be structured to facilitate analyses of alternative downstream decisions.

Parameter estimation (refer to the Taskforce parameter estimation paper for general recommendations)
Health-related DES models can incorporate a range of alternative parameter types, representing disease progression, clinical and administrative decision making algorithms, resource costs and constraints, health condition costs, and quality of life weights. Disease progression parameters are commonly represented as time to event parameters, i.e. the parameter describes the likelihood of a subsequent event(s) occurring at different (often continuous) time points. Decision making algorithms describe decisions made by clinicians with respect to treatment options and priority ordering of patients, and administrators with respect to implementing clinical priority orders. Resource costs and constraints refer to the costs associated with events that involve the use of specific resources, and their availability. Health condition costs and quality of life weights are attached to time spent in different health conditions to estimate long-term costs and health outcomes (i.e. QALYs).

Trade-off between structure and parameter estimation...

DES models facilitate complex model structures, and hence, they often require extensive data to populate them. In the absence of empirical data for some parameters, there are several options for proceeding. The most radical is to desist from building a model at all. This may be appropriate, depending on the purpose of the model, when there is extensive missing information, particularly on the core parameters (e.g. on the efficacy of treatment). Another option is to eliminate the sections of the model that require the parameters for which information is missing. This means restructuring the perspective of the model to the appropriate level of detail, for example, higher level models may provide insight into the problem and still have available valid data that a more micro model may not.

Alternatively, the original model structure may be maintained by deriving the missing parameter values via calibration, by identifying sets of input parameter values that produce output values that are similar to target values (i.e. observed estimates of the output parameters). This is a good option, but requires that suitable targets be available.

The use of expert elicited data as applied to DES...

In the absence of empirical data to estimate or calibrate input parameter values, such values may be elicited from experts working in the system being modelled. Expert elicitation is subject to a range of biases; both intentional and unintentional (e.g. recall bias). The strength (or value) of elicited parameter values will vary according to the complexity (or granularity) of the parameters for which values are being elicited, and the experience of the experts.
To increase confidence in values elicited from experts, it is important to validate their responses by asking additional questions in which elicited parameter values can be compared to empirical data. Another way is to cross check expert data from independent expert sources or use formal methods such as the Delphi method. In a DES, for example, clinicians might be asked to estimate the frequency of referrals to an allied health professional (a missing model parameter value), but also the frequency of surgical referrals (an empirically estimated model parameter). Their accuracy in estimating the latter provides some sense of their accuracy with respect to the unknown parameter.

**Recommendations**

*If parameter values are elicited from experts, formal elicitation methods should be used, and the elicited values should be validated.*

*If confidence in the elicited values is low, resulting analysis should be viewed only as a starting point for what-if analyses, and for estimating the value of collecting additional data.*

*If the decision is made to modify the originally specified model structure, the new structure must be carefully analysed to understand the potential effects of the omissions so as to inform decision makers of the additional uncertainty introduced. Explicit considerations of the size and likely direction of the effects of the modification should be presented.*

*Clinical guidelines are not always implemented...*

DES models often represent clinical and administrative decision making algorithms (e.g. processes for assigning patients to clinics). In the particular case of defining clinical algorithms, although clinical guidelines may specify resources that ought to be available and the decisions that clinicians ought to make, there is considerable evidence of variation in the uptake of guidelines.[7] Indeed, one of the purposes of the model may be to demonstrate the potential costs and benefits of adhering to published guidelines.

Baseline clinical and administrative algorithms (i.e. those applied in the current system) could be derived from analyses of patient records, though it is less taxing to ask clinicians and administrators what decisions they make, given specified circumstances.

**Recommendations:**

*It should not be assumed that relevant published guidelines are applied in the system being modelled.*
Ideally, clinical and administrative decision making algorithms should be based on analyses of observed decisions. If such a process is not feasible, algorithms should be developed with relevant personnel, and validated using routinely collected data, for example, extracting relevant data from a sample of patient records to compare observed and stated decisions.

Assigning times to next event...

When estimating the time to next event and there are two or more possible next events, alternative approaches to defining the next event include sampling:

a) separate times to each potential event, with the entity moving to the event with the earliest sampled time, or

b) an aggregate time to the next “event”, with a separate sampling process to determine the specific type of event that occurs (e.g. using multinomial regression analyses outputs to define the relevant probabilities). The event probabilities may vary as a function of the sampled time to the next event.

The first approach is more straightforward to parameterise: survival data for each event can be used directly, or parametric curves can be estimated for each event, and so it is easier to achieve a good fit between the predicted and observed data. The second approach uses a two stage process to estimate time to event parameters for each event and so it is more difficult to ensure a good fit between the observed and estimated event rates. The latter approach, however, better represents any correlation between time to event parameters, which will improve the accuracy of model sensitivity analyses.

Recommendation:

Where there are competing risks of events, parameterisation approaches that represent correlation between the likelihood of competing events are preferred to the specification of separate time to event curves for each event.

Representing continuous disease parameters...

In some cases, the likelihood of discrete events is a function of the value of a continuous measure (e.g. diabetic complications are a function of HbA1c, or clinical presentation is a function of tumour size) (as described in the model structure and design section). Time checks can be used to sample the likelihood of discrete events, conditional on the status of the continuous measure of disease progression (e.g. monthly time checks to update HbA1c levels and define related probabilities of complications).
Alternatively, it may be possible to define joint probability distributions that represent the combined likelihoods of disease progression and related events. In the diabetes example, we might sample the HbA1c level at which the first complication occurs, and then sample the time at which a patient reaches that level. This latter approach maintains a key asset of DES, namely that time moves forward when the next event occurs, not in fixed time cycles.

Recommendation:

Where possible, progression of continuous disease parameters and the likelihood of related events should be defined jointly to maintain the discrete event nature of DES, for example, sample the level of the continuous measure (e.g. HbA1c score) at which an event occurs, and then sample the time at which the level is reached.

Model implementation

Model implementation describes the process of transferring a defined model structure into a computer –based program, which can be populated and analysed. DES models generally represent complex systems, and their implementation requires some form of programming. Thus, implementation objectives are to ensure that the model structure is correctly implemented in a manner that promotes transparency and efficient analysis of the model.

A DES model can typically consist of the parts: Read Data, Create Patients, Main Section, Remove Patients and Print or Present results. The Main Section is the part containing the logic for the events, potential queuing and resource utilization, risk updating and everything else needed to happen between the beginning and end.

Consider using sub-models...

When planning implementation, the use of sub-models facilitates transparency by grouping related model logic (code) and presenting it to the reviewer in smaller chunks. Examples of sub-models include departments within a hospital (where the full system is the whole hospital); or the course of specific events such as hospitalization with myocardial infarction, stroke, bypass surgery, etc. (where the full system is the course of cardiovascular disease in a defined patient population).

Sub-models also mean less code, making the model easier to debug (each sub-model can be tested separately and identical code does not need to be verified in multiple instances), and therefore result in fewer errors. They also ensure that changes to the model will be consistently implemented for all strategies, and allow the model to be more readily updated as new information becomes available.
Recommendation:

To simplify debugging and updating, sub-models should be used to structure the model. When comparing two or more strategies within the same system (e.g. for the same condition in an HTA model), sub-models common to all strategies (e.g. progression following disease recurrence) should be defined once, and called from each strategy (i.e. all patients experiencing a recurrence pass through the common disease recurrence module).

Defining multiple model structures...

In some projects, there may be uncertainty around the defined model structure, which warrants the implementation and analysis of alternative model structures (i.e. structural sensitivity analysis). Rather than implementing separate models for each structure, the implementation of alternative model structures within a single DES reduces the probability of programming errors, as common code (or sub-models) can be referenced by multiple model structures. The use of a single model can also reduce the nuisance variance across model structures, e.g. through the use of common random numbers for shared sub-models (see model analysis section for more details).

Recommendation:

For structural sensitivity analyses, alternative structures should be implemented within a single DES.

Avoiding blocking events...

A common implementation error is to inadvertently block the possibility of events occurring, for example, patients at risk of stroke may have these risks ‘suspended’ whilst receiving hospital care following an ‘admission for myocardial infarction’ event. A parallel example in tree models is that it is good practice to put the risk of death branch at the beginning of the cycle to avoid this error.

Recommendation:

Analysts should ensure that the mechanism for applying ongoing risk(s) over multiple events remains active over the relevant time horizon.

Only collect outputs that required...

The manner in which a model is implemented determines the range and level of model outputs that can be used in the validation and final analysis of the model. As examples, if we are interested in the distribution of waiting times or costs across individual patients, it is necessary to implement the model so that each patient holds a record of their own waiting times and costs. If such detailed individual data
is not required, model complexity can be reduced by using global variables to inform mean parameter estimates across each model run.

**Recommendation:**

*Model implementation should account for the outputs required for the validation and the final analyses of the model. Where individual level data is required, relevant outputs should be stored as attributes, otherwise aggregated values should be collected from each model run to reduce the simulation burden.*

*General programming or dedicated DES software...*

Most DES models are implemented using either a general programming language or software developed specifically for the implementation of DES.

The advantages of using a general programming language are increased flexibility, faster execution and less dependence on proprietary software (apart from the programming language). The disadvantages are the need to write code for common basic functions (e.g. to administer the event list, run the queues, manage the resources, etc.), the more complex and extensive debugging required, and the lack of transparency. For many general programming languages, there are code libraries available to assist with many of the DES house-keeping tasks; and these can significantly improve coding efficiency and debugging.

Dedicated DES software is designed to overcome the limitations of general programming languages. They typically offer an attractive, easy-to-use interface that provides most of the required functions as modules that the modeller can readily incorporate in the model and much of the code required to implement the model is integrated with the modules. Time, event lists and other basic tasks are taken care of automatically behind the scenes. Commonly employed and understood graphical user interface aspects are utilized to make interaction with the model easier and more transparent and many of the software incorporate animation, which renders the model more visual and understandable. These features also facilitate debugging and greatly increase programming efficiency — the trade-off is somewhat reduced flexibility and calculation speed.

Many users of models, and even some modelers, prefer to use spreadsheets because they are perceived to be widely understood and it is felt that this increases transparency. However, spreadsheets are a poor tool for implementing simulation. The basic idea of a spreadsheet is to carry out all calculations simultaneously, thus implementing the sequential nature of calculations of a DES is very awkward. By the same token, it is difficult to mimic the continuous time component, which is one of the key-features
of DES. Moreover, spreadsheets rapidly grow in complexity and consequent lack of transparency; they offer the analyst few ready-built tools for creating, running or displaying a DES; and the use of programming in an accompanying language (such as VBA) defeats the purpose of building the model in the widely understood spreadsheet format.

**Recommendation:**

*The choice between using general programming or dedicated DES (“off-the-shelf”) software should be informed by the relative importance of flexibility and execution speed (general programming languages) vs. modelling efficiency, automated structure and transparency (dedicated DES software). Spreadsheet software is generally inappropriate for implementing DES and should not be used without justification.*

**Model analysis**

Within a DES model, entities (e.g. patients) are commonly subject to probability distributions that describe the cumulative probability of an event occurring at different times, for example, 0.05 by 28 days, 0.1 by 60 days, etc. Patients’ pathways through the model are determined by randomly sampling values from uniform distributions between 0 and 1, which are matched to the defined probability distributions. In the example, if a patient sampled a value of 0.1, they would experience the next event on day 60.

To estimate the model outputs associated with a single set of input parameter values, a DES undertakes a run that can be defined with respect to the number of entities that enter the system, and/or the time horizon of the model. Outputs of interest for each run may include mean values, calculated across all entities within the system, or the distribution of output parameter values across the entities.

The distribution of output parameter values may be of interest when evaluating systems such as an emergency department, to estimate the proportion of patients who have to wait more than a certain number of hours before being seen. Mean values are commonly of interest for HTA-based applications, where we are typically interested in the mean costs and outcomes because the question we are trying to answer is which technology provides the most cost-effective option across a defined patient group.

In both cases, even when using the same input parameter values, we will get different output values for every model run due to sampling variation. In both cases, we want the most accurate estimate of the mean values or the distributions of the output parameters, which can be achieved by either running more entities through the model and/or increasing the model’s time horizon.
In systems where the number of entities are sampled (e.g. patients presenting at an ED) and the time horizon is fixed (e.g. daily operation of an ED), we might undertake multiple replications of the model using the same input parameter values. From each replication, we might estimate outputs such as the proportion of days on which a certain percentage of patients waited longer than a threshold waiting time. Undertaking more replications will provide a tighter estimate of that output. One model run can consist of multiple replications.

Reducing run sizes...

The required size of each model run can be reduced by minimising the unwanted differences between alternative scenarios being evaluated (i.e. technologies or service configurations).[1] A good starting point is to use the same population for each scenario being evaluated, which ensures that, at least at the start, there are no unwarranted differences between the compared populations. Despite a common population at the beginning, nuisance variance can be introduced by the populations experiencing different pathways governed by the selection of random numbers. The application of separate streams of common random numbers to different events (e.g. one stream for sampling the occurrence of an MI, another for sampling length of stay, etc.) helps reduce the possibility that different random numbers are selected in each scenario. Variance reduction using common random numbers is additionally useful in software debugging, for example, the analyst can double check that the time to an event not influenced by the scenario is the same for the same simulated patient across strategies. Other more sophisticated techniques, such as signalling between populations to resynchronize their experience, can be implemented, thus further reducing required computing time.

Recommendations:

Procedures to lessen the impact of variation caused by the random processes (“variance reduction” techniques) should be implemented in all DES.

The number of entities in a single model run, duration of a model replication, or number of model replications in a run (using the same input parameter values) should be informed by the variance in the key model outputs across multiple model replications. It is good practice to simulate only the minimum number of entities required to provide sufficient confidence in the differences between strategies to inform decisions.

Probabilistic sensitivity analysis and model running times...
Handling uncertainty around the values of the input parameters is an additional process, which can be represented by undertaking a series of model runs using alternative sets of input parameter values (either deterministically or probabilistically – see the parameter estimation and uncertainty taskforce paper). Running times for probabilistic sensitivity analysis (PSA) can be large due to the combined requirement to reduce the variance around each run’s outputs and to undertake multiple runs. Rather than abandon PSA altogether, formulae based on ANOVA can be used to estimate the combined run size and number of model runs required to optimise the precision of the outputs, given an available (or desired) analytic time.[8]

**Recommendation:**

When run times for probabilistic sensitivity analysis are constrained, the optimal combination of run size (per input parameter set) and numbers of alternative input parameter sets tested should be estimated empirically to optimise the precision of the outputs of interest.

**Factorial design and optimum seeking approaches...**

If a DES is intended to evaluate a continuum of options (e.g. the level of cholesterol above which an intervention might be utilised) or there are multiple dimensions to alternative options (e.g. multiple staffing options for many staff categories) then it can become infeasible to test every possible option (or factor). Factorial design is recommended when there are multiple dimensions to each factor and one can reasonably conceive of ‘high’ and ‘low’ (or ‘on’ and ‘off’) values for each factor defining a strategy.[9] The aim is to understand how the output is related to the multiple factors. Instead of using one-way sensitivity analysis on the k factors, the factorial approach runs the model with each factor at either its ‘high’ or ‘low’ level i.e. a total of 2^k model runs. This provides estimates of the main effect of each factor and also of interactions between factors. When k is large, this can become prohibitive (e.g. 2^15=32,768 runs), suggesting use of ‘fractional factorial design’, where only a subset of the 2^k design points are used.

Optimum seeking approaches are useful when the decision maker is mainly interested in identifying the optimal strategy across many possible options. Optimum seeking uses iterative algorithms to assess the model outputs for the current configuration of options relative to a previously analysed set of options, which in turn informs the next set of options to evaluate. This iterative process is continued until a specified ‘stopping rule’ is achieved (e.g. a specific number of iterations or some ‘tolerance level’ for improvement in output response between iterations). There are many methods that can be used to decide on the next configuration to evaluate, including moving a certain number of steps in the direction
where performance appears to be improving, and using random jumps to avoid local optima. Such approaches are standard in the field of Optimization and often save substantial analyst and computer time.

**Recommendation:**

*If the number of strategies to compare is large or there are many structural assumptions to test, then “factorial design” and optimum seeking approaches should be used.*

*Meta modelling...*

Meta-modelling involves running a DES with different input parameter configurations, and then using regression methods to obtain an equation estimating the outputs as a function of the model’s input parameters.[7] The selection of configurations to run can be informed by factorial design, and the meta-model can in turn be used to inform and speed up factorial design and optimum seeking approaches. Gaussian process emulators have been used in health economic simulation models and have the advantage that uncertainty in the output can be represented for configurations not within the evaluated set, which enables quicker computation of PSA and expected value of information estimates.

**Recommendation:**

*When computing time precludes adequate representation of uncertainty, meta-modelling (statistical representation of the model input-output relationship) should be used.*

*Defining the starting state...*

A separate model analysis issue concerns the specification of the start state of the model. In many decision problems, the analysis does not start with a clean slate, for example, when simulating a hospital clinic that has been running for a number of years, the relevant starting point will be the current operation of the clinic, incorporating the cohort of patients who are currently booked or waiting to be booked into the clinic.

One option is to pre-load entities with existing attributes and history of events and start collecting results for analysis immediately. Pre-loading is appropriate if based on an integrated empirical dataset describing the current status of entities across the system being modelled.

The alternative is to run the model for some time prior to starting the main analysis - a warm-up period. From being empty at the beginning of the warm-up period, the system is built up to the current state
gradually, on the basis of input parameter values that will continue to be applied within the main
analysis phase of the model.

An important advantage of using a warm-up period is that it provides a form of validation of the model
by testing whether the populated model is able to create realistic starting conditions. However, the
process of matching current conditions can be difficult if input parameter values have changed over
time, for example, as a result of shifting referral patterns, or the introduction of new health
technologies. In such cases, the application of constant parameter values to predict current conditions
will misrepresent the parameter values to be applied within the main analysis phase of the model.

**Recommendation:**

*If the system to be modelled is not empty at the start of the time horizon to be evaluated, a warm-up
period should be used build the system up to the starting point if:*

- it can be reasonably assumed that the key parameters have remained constant over time, or
- the history of the key parameters can be incorporated into the warm-up period (e.g. the
  introduction of new health technologies can be described).

Otherwise, creating starting entities with ready-made histories (‘pre-loading’) is an acceptable approach.

**Representing and reporting DES models**

**Animation...**

As noted in the model implementation section, dedicated DES software often facilitates the animated
representation of models, where the key events are displayed and the passage of entities between the
events over the model time horizon is represented. Human beings are better at recognizing pattern and
problematic movement visually than via analysis of equations or data. Animation plays to this strength,
ensuring the identification of illogical movements in the model. It also provides a form of face validity,
where content experts can review the structure of the model and the movement of entities.

**Recommendation:**

*Animated representation of DES that displays the experience of events by individuals is recommended as
a means of engaging with users, as well to helping to debug the model through the identification of
illogical movements.*

**Diagrams...**
Reports of DES models should include diagrams that help the reader understand the model structure and function. Flow diagrams or State charts provide general frameworks for representing the key elements of a model, including the possible pathways between events (logic and causal relationships), and the presence of queues and decision points.

More detailed representations of the model structure should enable the reader to replicate the model structure (if they so wish). Module or event documentation figures can be used to describe the actions undertaken before, during, and after each event within the model. Lists of variables and attributes that are used or updated at each event are listed to provide the user with a detailed understanding on the underlying model process.

**Recommendation:**

*Both general and detailed representations of a DES model’s structure and logic should be reported to cover the needs of alternative users of the model. Detailed event documentation figures are also of benefit to the analyst, as a point of referral when returning to a model after a period of absence.*

**Conclusions**

DES provides flexible framework that can be used to model a wide variety of health care problems. Since it facilitates the representation of complex systems, there are a range of modelling issues along the model development, implementation, analysis, and reporting spectrum that should be addressed in order to maximise the value of the final model and its associated outputs. This paper has reviewed the main components of the modelling process and provided recommendations that should, if followed, increase the applicability, transparency, and value of DES models applied in health care context.
References


**Recommendations**

*Health-related outcomes are important...*

Constrained resource models should consider health-related outcomes, and not focus solely on measures of throughput. If not represented, their omission should be justified.

*The need to model constrained resources should be carefully considered...*

The effects of constrained resources should be modelled if:

- evaluated technologies result in differing levels of access (e.g. different referral rates), and
- time to access referred events can have significant effects on costs and/or outcomes (e.g. surgery)

*Downstream decisions...*

If downstream decisions can have significant effects on the differences in costs and/or outcomes of the primary object of an evaluation, the model should be structured to facilitate analyses of alternative downstream decisions.

*Trade-off between structure and parameter estimation...*

If parameter values are elicited from experts, formal elicitation methods should be used, and the elicited values should be validated.

If confidence in the elicited values is low, resulting analysis should be viewed only as a starting point for what-if analyses, and for estimating the value of collecting additional data.

If the decision is made to modify the originally specified model structure, the new structure must be carefully analysed to understand the potential effects of the omissions so as to inform decision makers of the additional uncertainty introduced. Explicit considerations of the size and likely direction of the effects of the modification should be presented.

*Clinical guidelines are not always implemented...*

It should not be assumed that relevant published guidelines are applied in the system being modelled.

Ideally, clinical and administrative decision making algorithms should be based on analyses of observed decisions. If such a process is not feasible, algorithms should be developed with relevant personnel, and validated using routinely collected data, for example, extracting relevant data from a sample of patient records to compare observed and stated decisions.
Assigning times to next event...

Where there are competing risks of events, parameterisation approaches that represent correlation between the likelihood of competing events are preferred to the specification of separate time to event curves for each event.

Representing continuous disease parameters...

Where possible, progression of continuous disease parameters and the likelihood of related events should be defined jointly to maintain the discrete event nature of DES, for example, sample the level of the continuous measure (e.g. HbA1c score) at which an event occurs, and then sample the time at which the level is reached.

Consider using sub-models...

To simplify debugging and updating, sub-models should be used to structure the model. When comparing two or more strategies within the same system (e.g. for the same condition in an HTA model), sub-models common to all strategies (e.g. progression following disease recurrence) should be defined once, and called from each strategy (i.e. all patients experiencing a recurrence pass through the common disease recurrence module).

Defining multiple model structures...

For structural sensitivity analyses, alternative structures should be implemented within a single DES.

Avoiding blocking events...

Analysts should ensure that the mechanism for applying ongoing risk(s) over multiple events remains active over the relevant time horizon.

Only collect outputs that required...

Model implementation should account for the outputs required for the validation and the final analyses of the model. Where individual level data is required, relevant outputs should be stored as attributes, otherwise aggregated values should be collected from each model run to reduce the simulation burden.

General programming or dedicated DES software...

The choice between using general programming or dedicated DES (“off-the-shelf”) software should be informed by the relative importance of flexibility and execution speed (general programming languages)
vs. modelling efficiency, automated structure and transparency (dedicated DES software). Spreadsheet software is generally inappropriate for implementing DES and should not be used without justification.
Reducing run sizes...

Procedures to lessen the impact of variation caused by the random processes ("variance reduction" techniques) should be implemented in all DES.

The number of entities in a single model run, duration of a model replication, or number of model replications in a run (using the same input parameter values) should be informed by the variance in the key model outputs across multiple model replications. It is good practice to simulate only the minimum number of entities required to provide sufficient confidence in the differences between strategies to inform decisions.

Probabilistic sensitivity analysis and model running times...

When run times for probabilistic sensitivity analysis are constrained, the optimal combination of run size (per input parameter set) and numbers of alternative input parameter sets tested should be estimated empirically to optimise the precision of the outputs of interest.

Factorial design and optimum seeking approaches...

If the number of strategies to compare is large or there are many structural assumptions to test, then “factorial design” and optimum seeking approaches should be used.

Meta modelling...

When computing time precludes adequate representation of uncertainty, meta-modelling (statistical representation of the model input-output relationship) should be used.

Defining the starting state...

If the system to be modelled is not empty at the start of the time horizon to be evaluated, a warm-up period should be used build the system up to the starting point if:

- it can be reasonably assumed that the key parameters have remained constant over time, or
- the history of the key parameters can be incorporated into the warm-up period (e.g. the introduction of new health technologies can be described).

Otherwise, creating starting entities with ready-made histories (‘pre-loading’) is an acceptable approach.

Animation...
Animated representation of DES that displays the experience of events by individuals is recommended as a means of engaging with users, as well to helping to debug the model through the identification of illogical movements.

Diagrams...

Both general and detailed representations of a DES model’s structure and logic should be reported to cover the needs of alternative users of the model. Detailed event documentation figures are also of benefit to the analyst, as a point of referral when returning to a model after a period of absence.